Term project 최종 발표

Data Mining을 이용한 서울대공원 2019년 4월 입장객 수 예측

동물원은 서울대공원이조

발표자: 201701071 김하나

201703317 정혜원

201600961 김형인

0. 목차

01. Project 개요

02. 보완 내용

03. 데이터

04. 데이터 분석

05. 결론

06. 느낀점

07. Q&A

1. Project 개요
(1) 연구 배경 및 목표

구분	전주동물원	광주우치 동물원	청주동물원	대전오월드 동물원	어린이대공원 동물원	서울대공원 동물원
위치	전주시 덕진구	광주광역시 북구	충청북도 청주시	대전광역시 중구	서울시 광진구	경기도 과천시
면적	125,380m²	121,302m2	126,900m2	약 130,000m2	30,054m2	**196,000m2**
입장객수	749,008명 (2010년)	767,630명 (2010년)	309,285명 (2007년)	약1,000,000명 (2009년)	12,922,469명 (2010년)	**2,851,974명 (2010년)**
동물종수	107종 665마리	66종 264마리	125종 557마리	120종 584마리	92종 542마리	**307종 2757마리**

→ 국내 최대 규모의 단일 동물원 & 다양하고 많은 동물의 종

동물원은 야외에 있으므로 날씨에 구애 받고, 주말에 더 북적

∴ 입장객 수의 증감에 영향을 미치는 요인들이 무엇인지 파악 & 입장객 수 예측

→ 서울대공원: 입장객 수에 대비한 적절한 수의 직원 배치

입장객: 입장객 수를 통한 방문계획 수립 가능

1. Project 개요
(2) 연구 방향

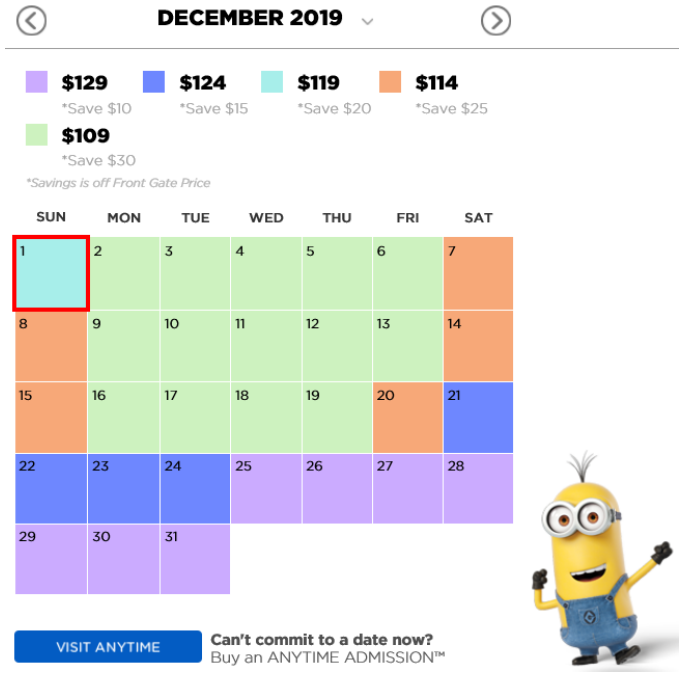
1. 브레인스토밍을 통한 **요인 선정**

2. **탐색적 데이터 분석** → SPSS Modeler를 통한 그래프 작성, NA값 & 이상치 확인

3. R3.6.1을 통한 **전처리** 과정

4. **시계열 분석, 로지스틱회귀, 의사결정 나무**를 통한 예측

5. 결론 도출 및 2019년 4월 입장객 수 **예측 표 시각화**



2. 보완 내용

[제안서 보완내용]

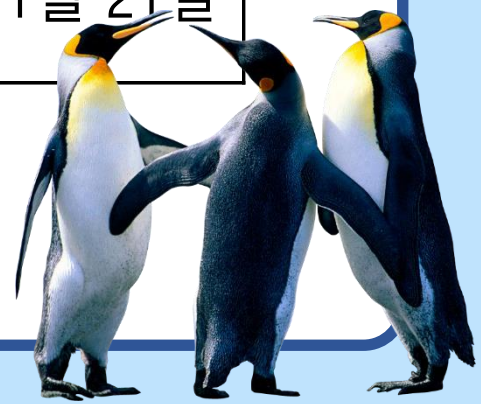
1. **연구배경 수정**
2. **데이터 삭제** - 질병 데이터, 하늘 상태 데이터
3. **데이터 추가** – 서울대공원 **행사 데이터**, 서울/경기 지역의 **온도 데이터**, 서울/경기남부 **미세먼지**, **초미세먼지** 데이터
4. **분석방법 변경** – 시계열 분석, 의사결정 분석

[중간보고서 보완내용]

1. 보고서에 행사 데이터 엑셀 파일 일부 첨부
2. **이상치 제거 및 변경**
3. 시계열분석 방법 재정립
4. 일부 데이터 연속형 → **범주형 변수로 변경**
5. 예측된 일일 **입장객수 등급 시각화**

3. 데이터
(1) Raw 데이터 수집 경로

데이터	출처	날짜
서울대공원 입장객 수 데이터	서울 열린 데이터 광장	2009년 1월 1일~2019년 4월 30일
기온 데이터	기상자료개방포털	2009년 1월 1일~2019년 11월 18일
강수량 데이터	기상자료개방포털	2009년 1월 1일~2019년 11월 18일
미세먼지/초미세먼지 데이터	에어코리아	2015년 1월 1일~2019년 11월 3일
서울대공원 행사 데이터	서울시 120 다산콜센터 네이버 블로그	2009년 1월 1일~2019년 11월 21일



3. 데이터
(2) 데이터 목록

1. 서울대공원 입장객 데이터

- 데이터 개수

: 약 220,000개
(2011년 1월1일~1월 24일, 2014년 2월 전체, 2017년 1월 전체, 2월 전체, 3월 1일~3월 27일 없음)

- Attribute 목록

No	Atrribute		표현방식/단위
1	날짜	날짜	연속형
		요일	명목형
2	날씨	날씨	명목형
3	월별입장객	유료합계	연속형
		무료합계	
		총계	
		외국인계	
4	동,식물원	유료소계	연속형
		어른	
		청소년	
		어린이	
		외국인	
		단체입장	
		무료소계	
		총계	
5	돌고래쇼	유료소계	연속형
		어른	
		청소년	
		어린이	
		외국인	
		단체입장	
		무료소계	
		총계	
6	테마가든	유료소계	연속형
		어른	
		청소년	
		어린이	
		외국인	
		단체입장	
		무료소계	
		총계	
7	캠프입장	유료소계	연속형
		어른	
		청소년	
		어린이	
		외국인	
		단체입장	
		무료소계	
		총계	
8	자연학습교실	유료소계	연속형
		어른	
		청소년	
		어린이	
		외국인	
		단체입장	
		무료소계	
		총계	
9	기타	유료소계	연속형
		어른	
		청소년	
		어린이	
		외국인	
		단체입장	
		무료소계	
		총계	

3. 데이터
(2) 데이터 목록

2. 기온 데이터

- 데이터 개수

: 약 39,000개

- **Attribute 목록**

No	Attribute	표현방식/단위
1	지역번호	명목형
2	지역명	명목형
3	일시	연속형
4	평균기온(℃)	연속형
5	평균최고기온(℃)	연속형
6	최고기온(℃)	연속형
7	최고기온관측지점	명목형
8	평균최저기온(℃)	연속형
9	최저기온(℃)	연속형
10	최저기온관측지점	명목형

3. 강수량 데이터

- 데이터 개수

: 약 31,200개

- **Attribute 목록**

No	Attribute	표현방식/단위
1	지역번호	명목형
2	지역명	명목형
3	일시	연속형
4	평균일강수량(mm)	연속형
5	최다일강수량(mm)	연속형
6	최다강수량지점	명목형
7	1시간최다강수량(mm)	연속형
8	1시간최다강수량지점	명목형

3. 데이터
(2) 데이터 목록

4. 미세먼지/초미세먼지 데이터

- 데이터 개수

: 약 800개

- **Attribute 목록**

No	Attribute	표현방식/단위
1	지역	명목형
2	권역	명목형
3	항목	명목형
4	경보단계	명목형
5	발령시간	연속형
6	해제시간	연속형

5. 행사 데이터

- 데이터 개수

: 약 4,574개

- **Attribute 목록**

No	Attribute	표현방식/단위
1	날짜	연속형
2	동물원 행사 유무	이분형
3	테마파크 행사 유무	이분형
4	기타 행사 유무	이분형
5	총 행사 유무	이분형

- 행사 데이터 예시

event_date	event_zoo	event_theme	event_etc	event_total
2009-01-01	0	0	0	0
2009-01-02	0	0	1	1
2009-01-03	0	0	1	1
2009-01-04	0	0	1	1
2009-01-05	0	0	1	1
2009-01-06	0	0	1	1
2009-01-07	0	0	1	1
2009-01-08	0	0	1	1
2009-01-09	0	0	1	1
2009-01-10	0	0	1	1
2009-01-11	0	0	1	1
2009-01-12	0	0	1	1
2009-01-13	0	0	1	1
2009-01-14	0	0	1	1
2009-01-15	0	0	1	1
2009-01-16	0	0	1	1
2009-01-17	0	0	1	1

3. 데이터
(3) 탐색적 데이터 분석

1. 통계적 특성 파악

[데이터 정리]
- R3.6.1 이용
- 변수 생성 및 데이터 통합
- 각 데이터의 필요한 부분만 선택
- 적절하지 않은 변수명 변경
- 필요하지 않은 column 삭제

[데이터 살펴보기]
- 데이터의 head, tail, 차원,속성, 요약 통계량

```
> table(is.na(seoul_merge))

FALSE   TRUE
86132 13618
```

- 결측값 13,618개 → 전체의 13%

∴ NA값 데이터 삭제 X → 채워 넣는 방법 선택

```
> summary(seoul_merge)
      Date                Day                 AA               AY               AC               AF               AG            A_free            AT              TA               TY
 Min.   :2009-01-01 00:00:00   Length:3990   Min.   :    0   Min.   :   0.0   Min.   :    0.0   Min.   :   0.00   Min.   :    0.0   Min.   :    0   Min.   :    0   Min.   :   0.0   Min.   : -1.0
 1st Qu.:2011-09-25 06:00:00   Class :character   1st Qu.:  600   1st Qu.:  34.0   1st Qu.:   60.0   1st Qu.:   8.00   1st Qu.:    0.0   1st Qu.:  826   1st Qu.: 1758   1st Qu.:  12.0   1st Qu.:  0.0
 Median :2014-06-18 12:00:00   Mode  :character   Median : 1640   Median :  133.0   Median :  256.0   Median :  30.00   Median :  112.0   Median : 1332   Median : 4055   Median :  89.0   Median :  1.0
 Mean   :2014-06-16 01:46:06                      Mean   : 3964   Mean   :  516.1   Mean   :  820.8   Mean   :  70.26   Mean   :  958.4   Mean   : 1946   Mean   : 8071   Mean   : 521.4   Mean   : 19.7
 3rd Qu.:2017-03-08 18:00:00                      3rd Qu.: 3738   3rd Qu.:  438.0   3rd Qu.:  917.0   3rd Qu.:  77.00   3rd Qu.:  860.0   3rd Qu.: 2342   3rd Qu.:10259   3rd Qu.: 466.0   3rd Qu.:  7.0
 Max.   :2019-11-21 00:00:00                      Max.   :51350   Max.   :32744.0   Max.   :21658.0   Max.   :2369.00   Max.   :33399.0   Max.   :40401   Max.   :95153   Max.   :11348.0   Max.   :2909.0
                                                  NA's   :365     NA's   :365       NA's   :365       NA's   :365       NA's   :365       NA's   :365     NA's   :365     NA's   :365       NA's   :365
       TC               TF               TG             T_free            TT               ST            rain_mean        temper_mean        event_zoo       event_theme        event_etc         event_total
 Min.   :   0.00   Min.   :  0.000   Min.   : -20   Min.   :   0.0   Min.   :    0   Min.   :    0   Min.   :  0.000   Min.   :-14.60   Min.   :0.000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
 1st Qu.:   1.00   1st Qu.:  0.000   1st Qu.:   0   1st Qu.:  20.0   1st Qu.:   40   1st Qu.: 1870   1st Qu.:  0.000   1st Qu.:  3.50   1st Qu.:0.000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
 Median :   6.00   Median :  0.000   Median :   0   Median :  94.0   Median :  240   Median : 4628   Median :  0.000   Median : 14.00   Median :0.000   Median :0.00000   Median :0.00000   Median :1.0000
 Mean   :  75.88   Mean   :  9.317   Mean   :  107   Mean   : 330.9   Mean   : 1064   Mean   : 9135   Mean   :  3.453   Mean   : 12.54   Mean   :0.489   Mean   :0.09073   Mean   :0.02581   Mean   :0.5439
 3rd Qu.:  60.00   3rd Qu.:  6.000   3rd Qu.:  54   3rd Qu.: 416.0   3rd Qu.: 1286   3rd Qu.:11940   3rd Qu.:  0.600   3rd Qu.: 22.00   3rd Qu.:1.000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:1.0000
 Max.   :6740.00   Max.   :998.000   Max.   :7179   Max.   :5685.0   Max.   :22396   Max.   :101527   Max.   :201.600   Max.   : 31.90   Max.   :1.000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
 NA's   :365       NA's   :365       NA's   :365    NA's   :365      NA's   :365     NA's   :365      NA's   :3         NA's   :3
    dust           warninglevel
Length:3990        Length:3990
Class :character   Class :character
Mode  :character   Mode  :character
```

2. 결측값 처리 – NA값을 다른 값으로 채워 넣기

: 전년도와 이듬해 같은 날짜의 데이터 두 쌍을 평균을 내어 채움.

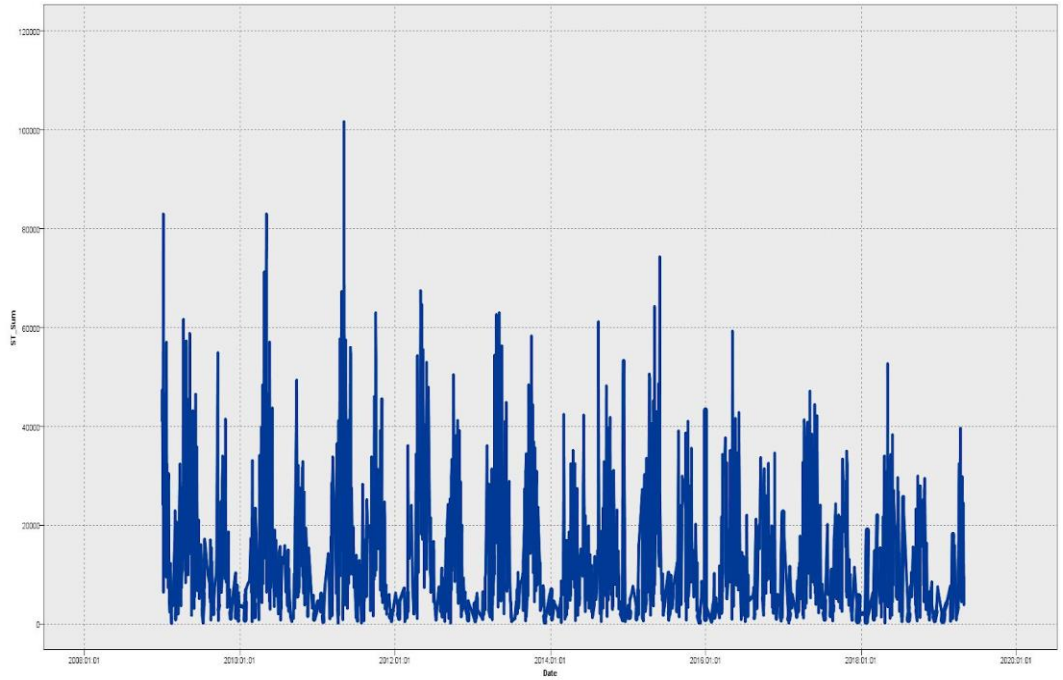
ex) 2011년 1월 4일 행의 4열~18열의 데이터 NA

: 2010년 1월 4일의 4열~18열 데이터 & 2012년 1월 4일의 4열~18열 데이터
각각의 평균값 도출

→ 2011년 1월 4일 행의 4열~18열 데이터에 채워 넣기

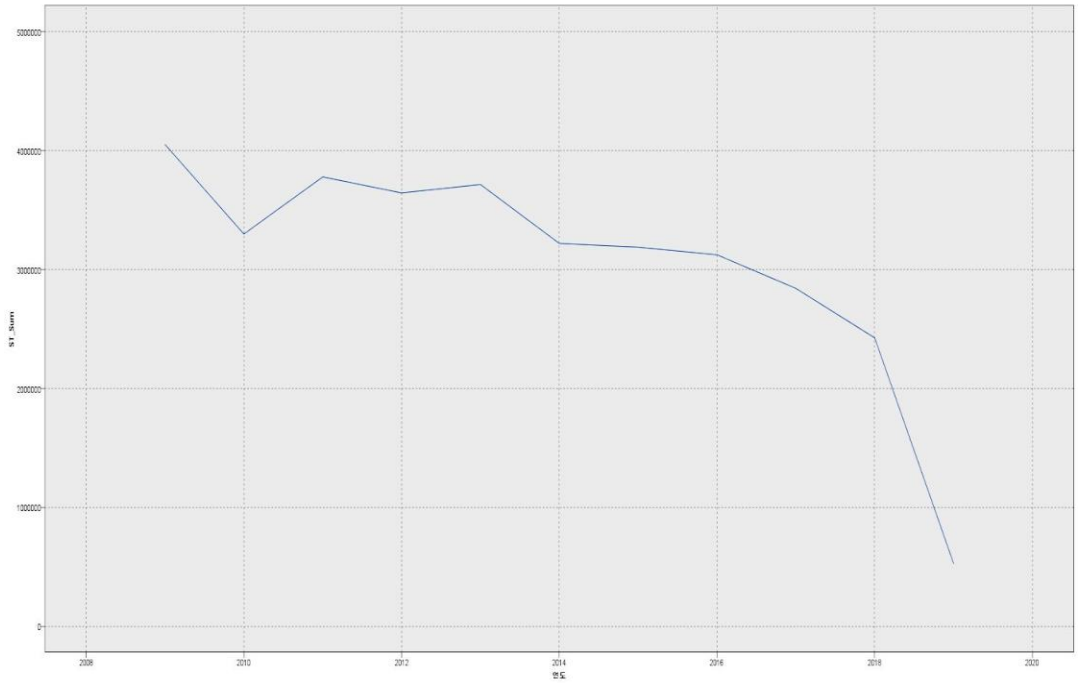
3. 그래프를 통한 데이터 탐색

- 일별 입장객 수 그래프



X축 : 일, Y축 : 입장객 수

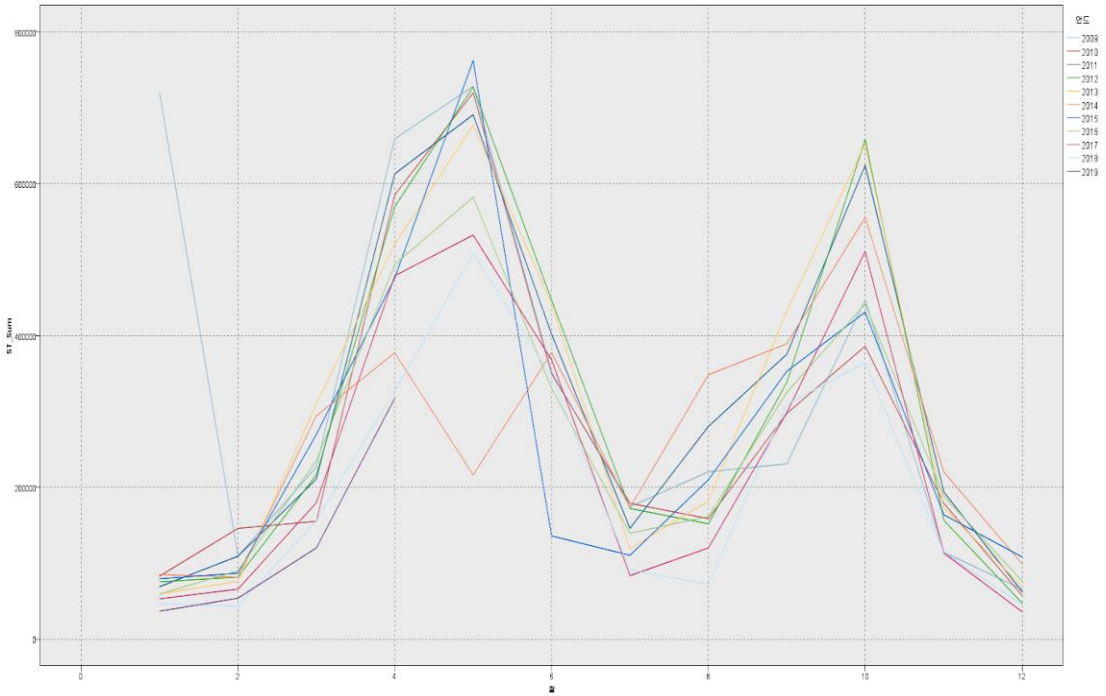
- 연도별 입장객 합계 그래프



X축 : 년, Y축 : 연도 별 입장객 수 합계

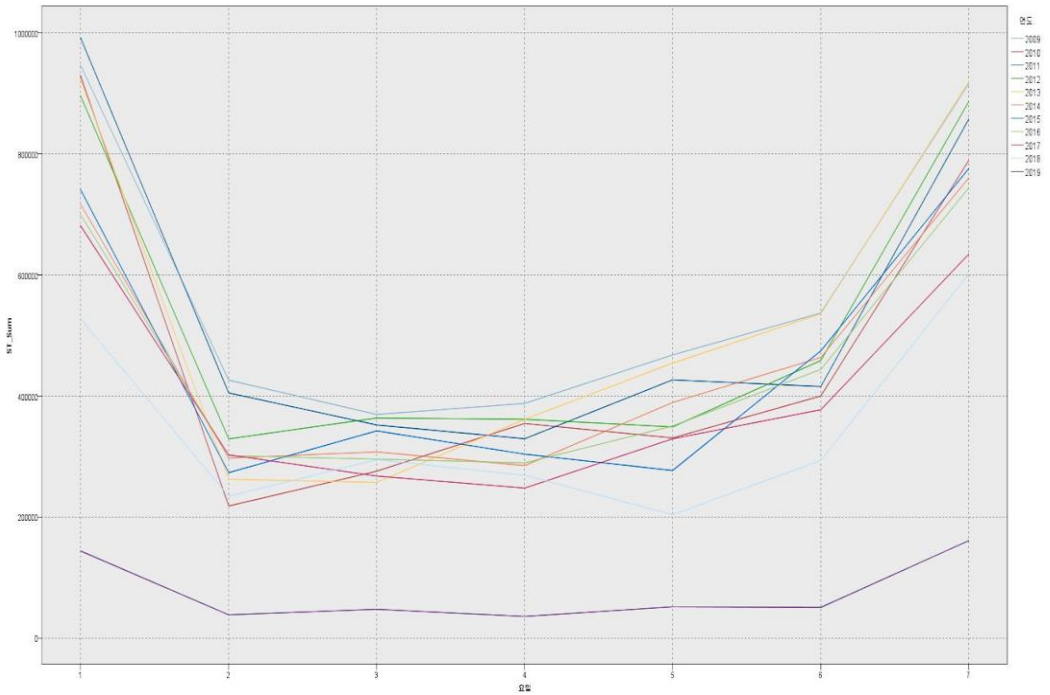
3. 그래프를 통한 데이터 탐색

- 월별 입장객 합계 그래프



X축 : 월, Y축 : 월별 입장객 수 합계

- 요일별 입장객 합계 그래프



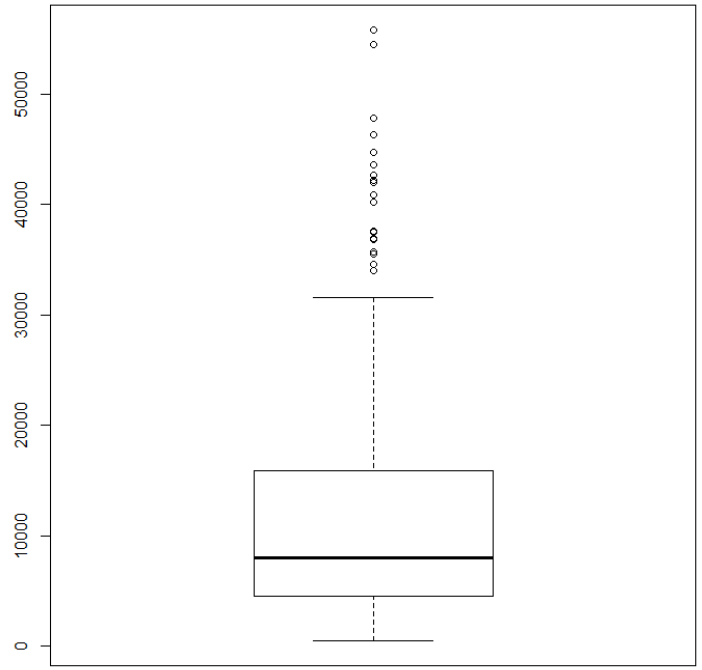
X축 : 요일, Y축 : 요일별 입장객 수 합계

→ 데이터 탐색 결과,
→ 눈에 띄는 이상치 제거 필요성

4. 이상치 제거

1. 최소값: 제 1사분위에서 1.5 IQR을 뺀 위치
2. 제 1사분위(Q1): 25% 위치
3. 제 2사분위(Q2): 50% 위치. 중앙값(median)
4. 제 3사분위(Q3): 75% 위치
5. 최대값: 제 3사분위 + 1.5 IQR

→ 최소값과 최대값을 넘어가는 위치의 값
: 이상치(Outlier)

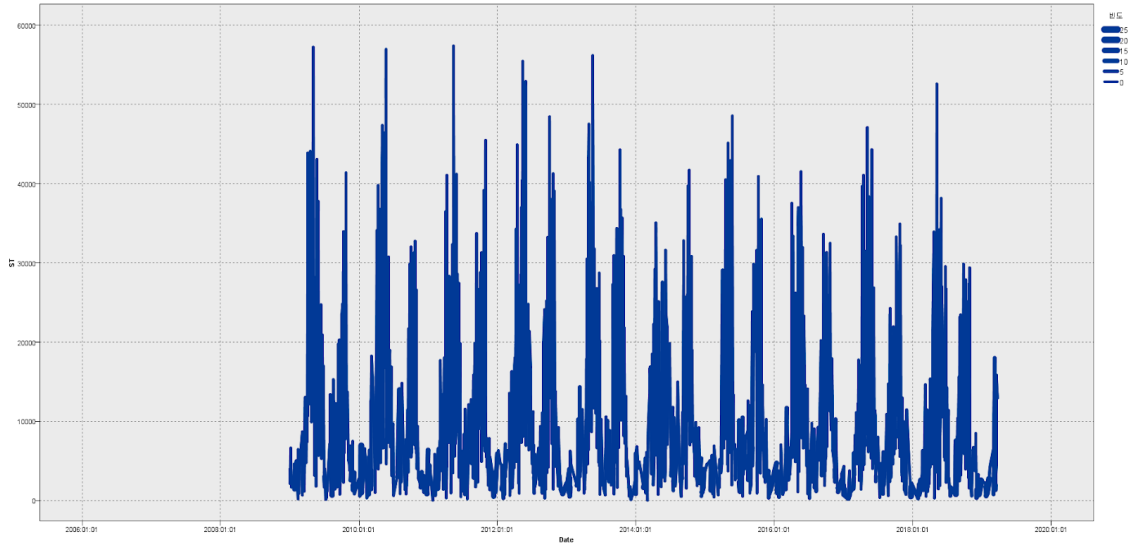



```
> #6월 이상치
> boxplot(jun)
> boxplot(jun)$stats
      [,1]
[1,]   471
[2,]  4501
[3,]  7974
[4,] 15901
[5,] 31539
attr(,"class")
        1
"integer"
> jun_clean <- ifelse(jun < 471 | jun >31539, NA , jun)
> table(is.na(jun_clean))

FALSE  TRUE
  281    19
```

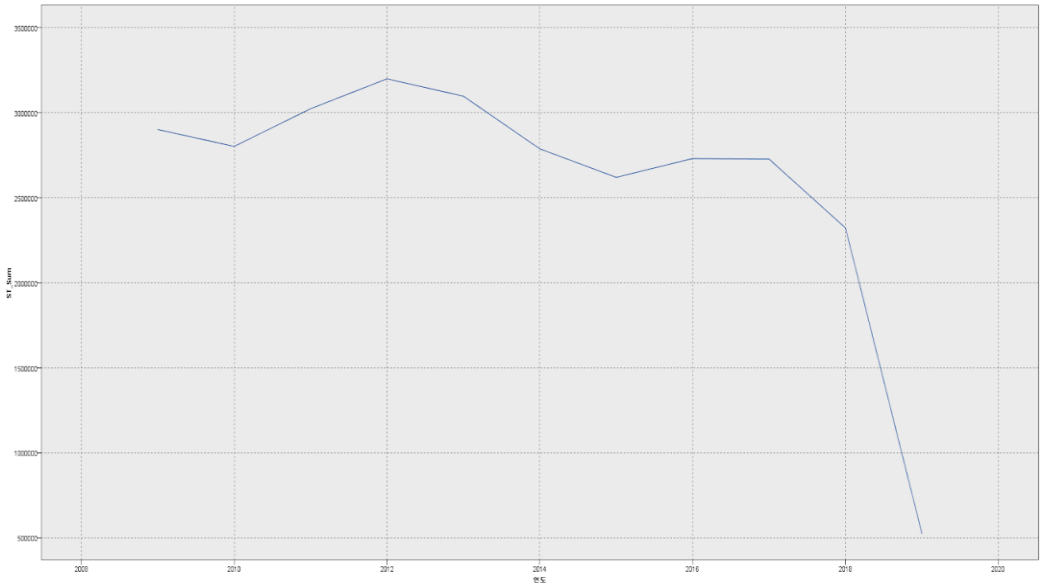
5. 이상치 제거 후 그래프를 통한 데이터 탐색

- 일별 입장객 수 그래프



X축 : 일, Y축 : 입장객 수

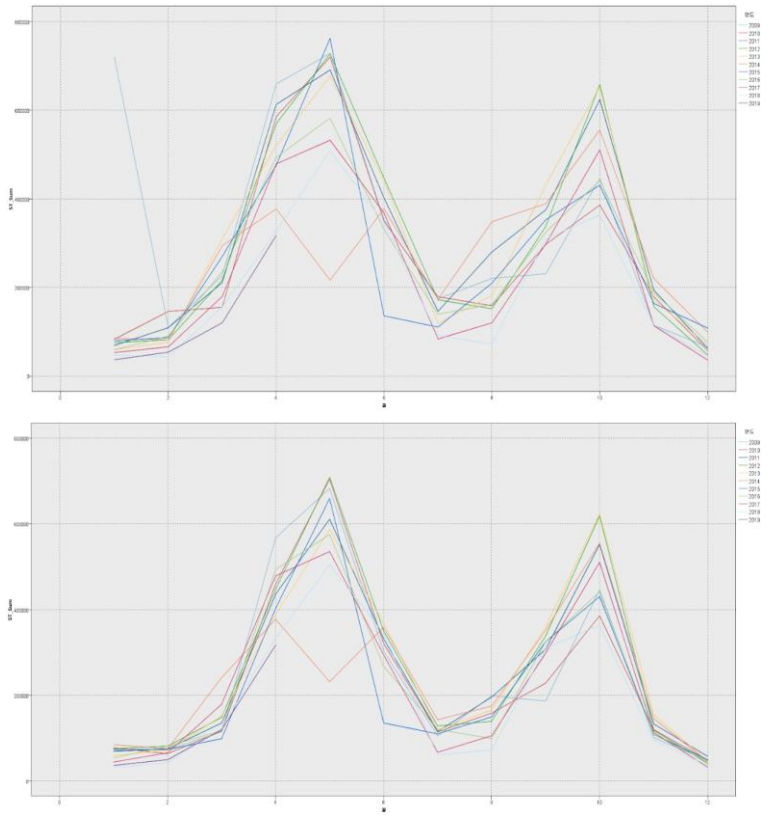
- 연도별 입장객 합계 그래프



X축 : 년, Y축 : 연도 별 입장객 수 합계

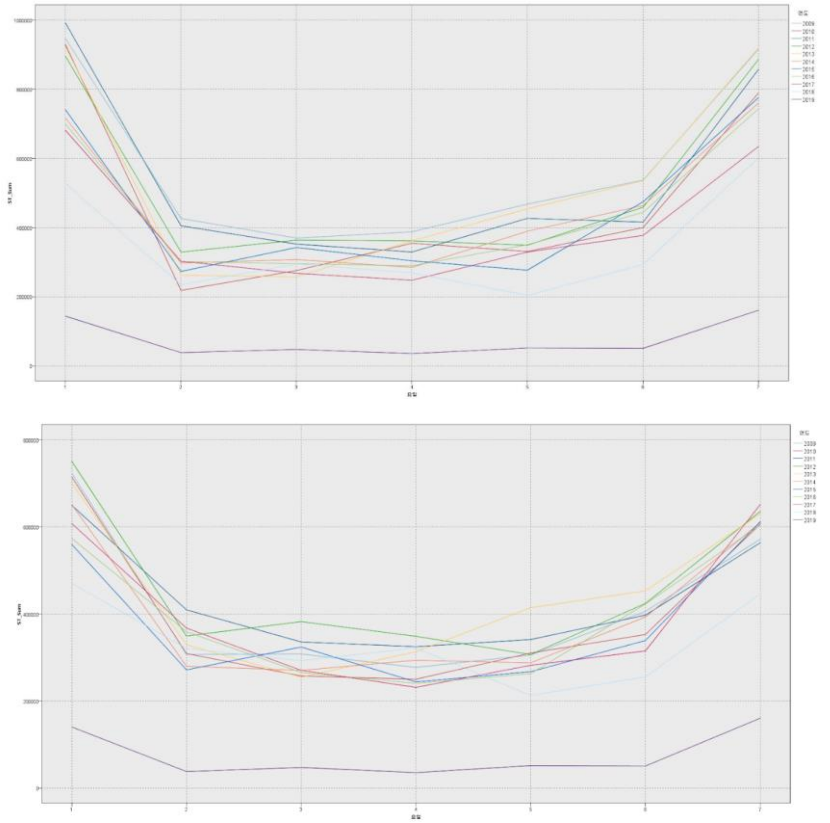
5. 이상치 제거 후 그래프를 통한 데이터 탐색

- 월별 입장객 합계 그래프



X축 : 월, Y축 : 월별 입장객 수 합계

- 요일별 입장객 합계 그래프



X축 : 요일, Y축 : 요일별 입장객 수 합계

- **데이터 변환**

입장객 수(서울대공원 입장객 데이터) 구간화

- 2009년~2019년 매 해의 입장객 최소값 & 최대값 계산

	min	max
2009	104	57166
2010	314	56911
2011	9	57315
2012	211	55415
2013	123	56161
2014	11	41707
2015	352	48511
2016	294	41495
2017	237	47050
2018	128	52583
2019	436	39526

→ min 평균: 244.3889 , max 평균: 50349

→ 평균들의 차를 통해 5개의 구간으로 구간화

A	B	C	D	E
~ 10314명	10315명 ~ 20384명	20385명 ~ 30454명	30455명 ~ 40524명	40524명 ~

(1) 월별예측

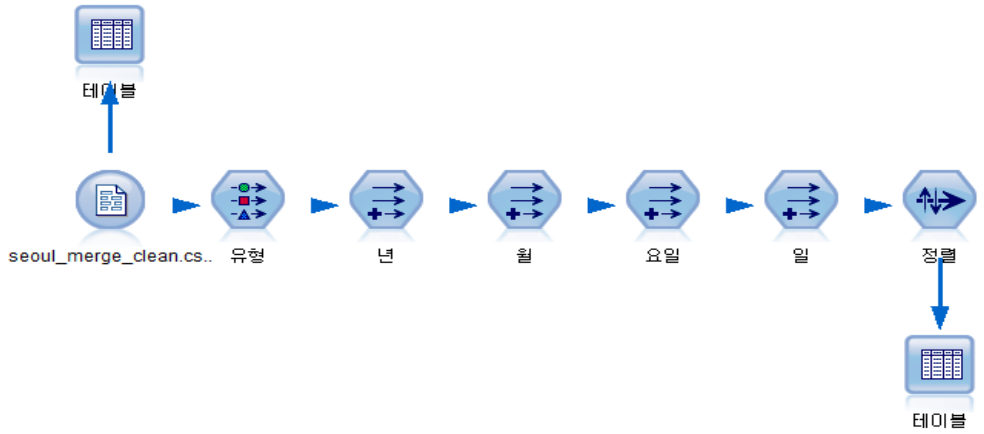
- 2009년 1월 1일~2019년 3월 31일의 데이터로 2019년 4월 총이용객 수 예측
- 2019년 4월의 기존 데이터와 비교

• 사용한 데이터 Attribute

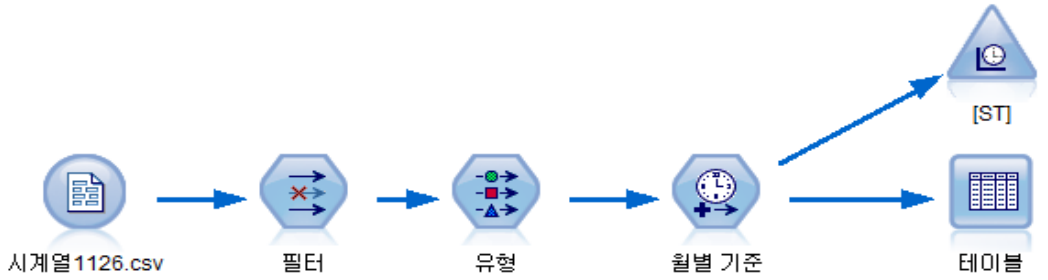
No	Attribute	Attribute 내용	표현방식/단위
1	Date	날짜	연속형
2	Day	요일	명목형
3	ST	일일 입장객수	연속형
4	rain_mean	강수량	연속형
5	temper_mean	기온	연속형
6	event_total	행사의 유무	이분형
7	dust	미세먼지/초미세먼지 유무	명목형
8	warninglevel	미세먼지/초미세먼지 수준	명목형

4. 데이터 분석
(1) 시계열 분석

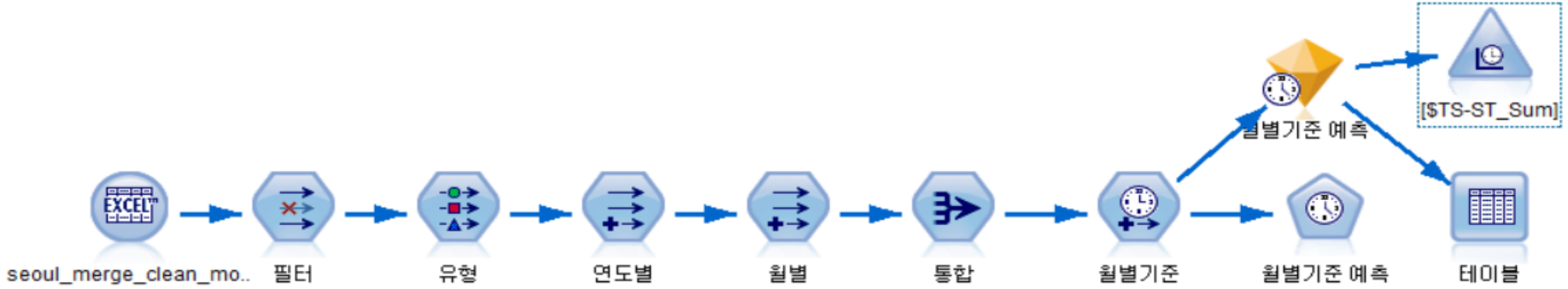
- 시계열 데이터.csv 파일 생성



- 기존 데이터 모형



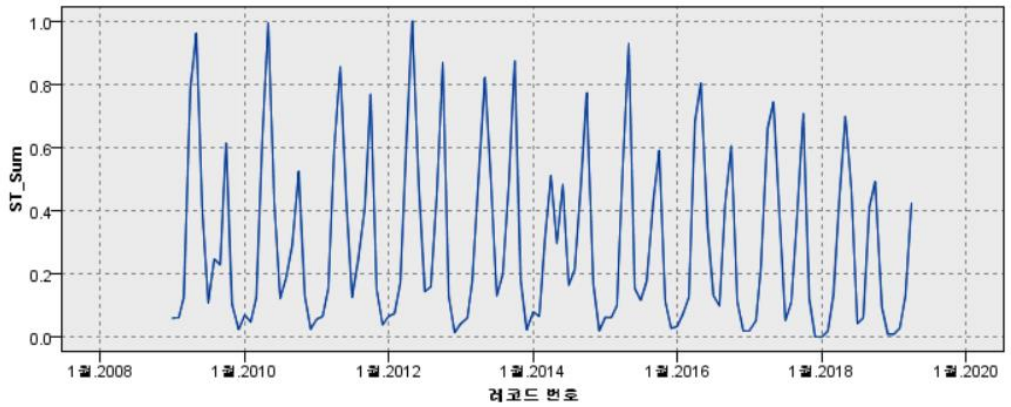
- 예측 데이터 모형



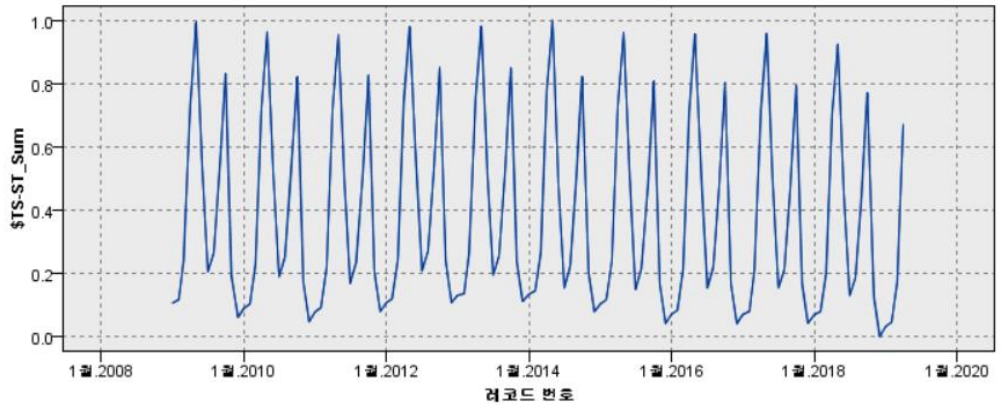
4. 데이터 분석
(1) 시계열 분석

- 월별 예측

1) 2019년 4월까지의 기존 분포



2) 2019년 4월 예측 분포



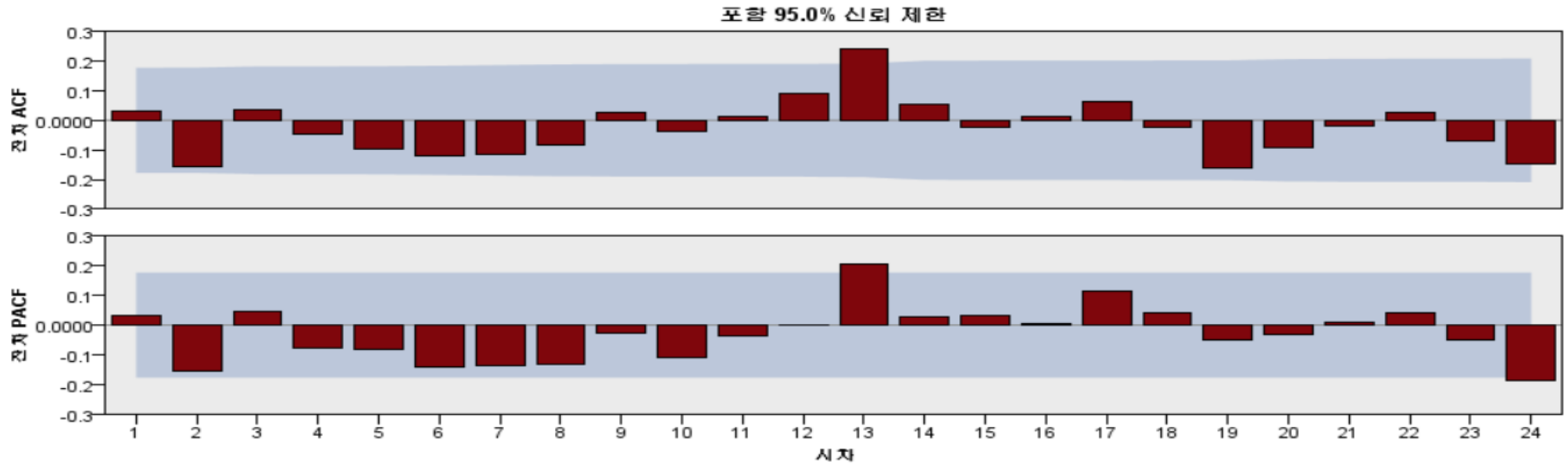
→ 월별 기존 값(ST_SUM) / 예측 값($TS-ST_SUM)

$TI_TimeLabel	ST_Sum
10월 2017	510691
11월 2017	113167
12월 2017	32187
1월 2018	30954
2월 2018	43635
3월 2018	123948
4월 2018	333942
5월 2018	504669
6월 2018	345196
7월 2018	60733
8월 2018	72347
9월 2018	309478
10월 2018	364014
11월 2018	95079
12월 2018	36768
1월 2019	36675
2월 2019	50214
3월 2019	120176
4월 2019	317093

$TI_TimeLabel	$TS-ST_Sum
10월 2017	481574
11월 2017	111430
12월 2017	36646
1월 2018	52955
2월 2018	57783
3월 2018	130555
4월 2018	425714
5월 2018	557822
6월 2018	280301
7월 2018	87582
8월 2018	122516
9월 2018	274073
10월 2018	467832
11월 2018	84449
12월 2018	10551
1월 2019	29915
2월 2019	37607
3월 2019	113043
4월 2019	408859

4. 데이터 분석
(1) 시계열 분석

3) 예측 결과에 대한 잔차



→ 자기상관함수(ACF)와 편자기상관함수(PACF) 이용하여 오차항들의 자기상관 확인 결과, 대체적으로 파란색 칸을 넘지 않아 자기상관이 없으므로 **좋은 모형**

4) 정확도 및 적합도 확인

목표	모델	예측변수	Stationary...	R**2	RMSE	MAPE	MAE
ST_Sum	단순 계절모델	0	0.731	0.902	58,663.976	19.604	35,588.626

(2) 일별예측

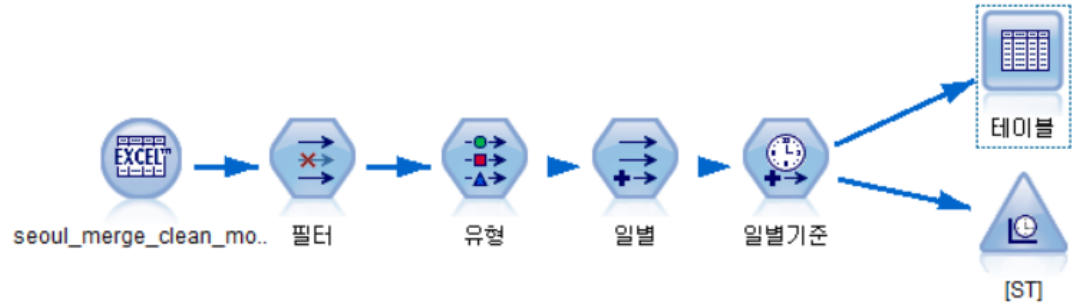
- 2009년 1월 1일~2019년 3월 31일의 데이터로 2019년 4월 일일 이용객 수 예측
- 2019년 4월의 기존 데이터와 비교

• 사용한 데이터 Attribute

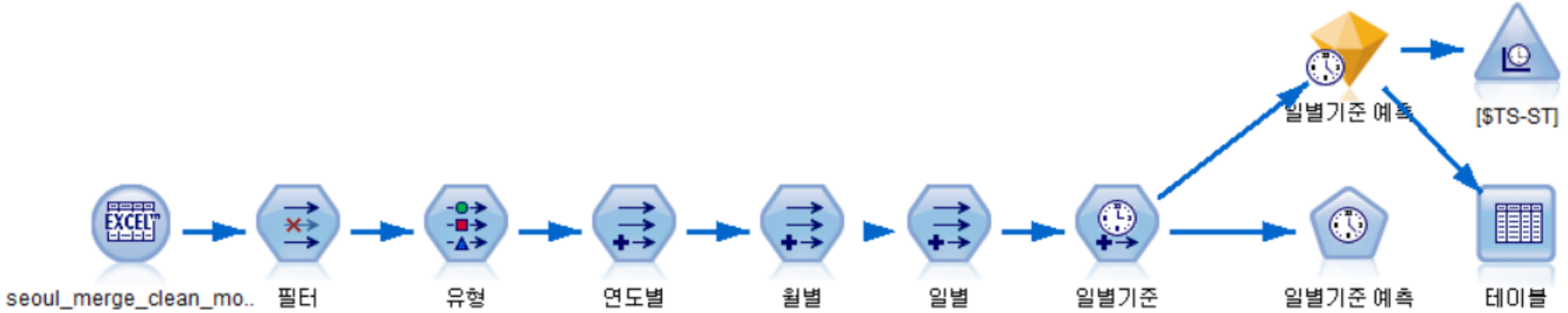
No	Attribute	Attribute 내용	표현방식/단위
1	Date	날짜	연속형
2	Day	요일	명목형
3	ST	일일 입장객수	연속형
4	rain_mean	강수량	연속형
5	temper_mean	기온	연속형
6	event_total	행사의 유무	이분형
7	dust	미세먼지/초미세먼지 유무	명목형
8	warninglevel	미세먼지/초미세먼지 수준	명목형

4. 데이터 분석
(1) 시계열 분석

- 기존 데이터 모형



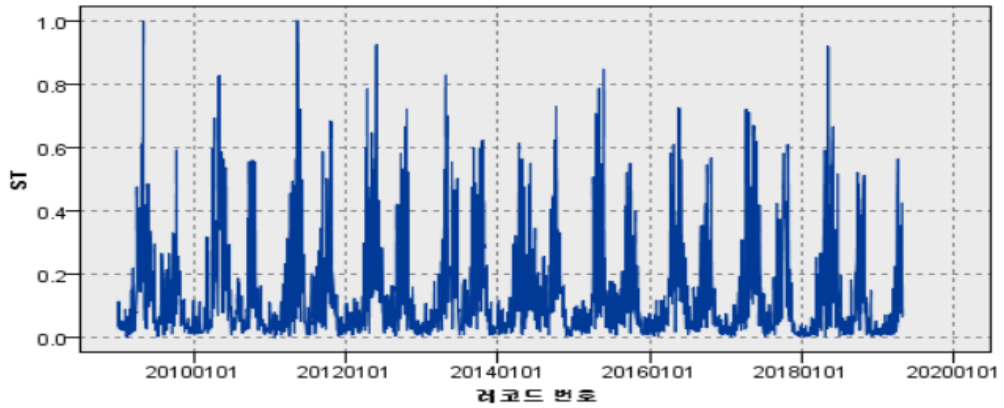
- 예측 데이터 모형



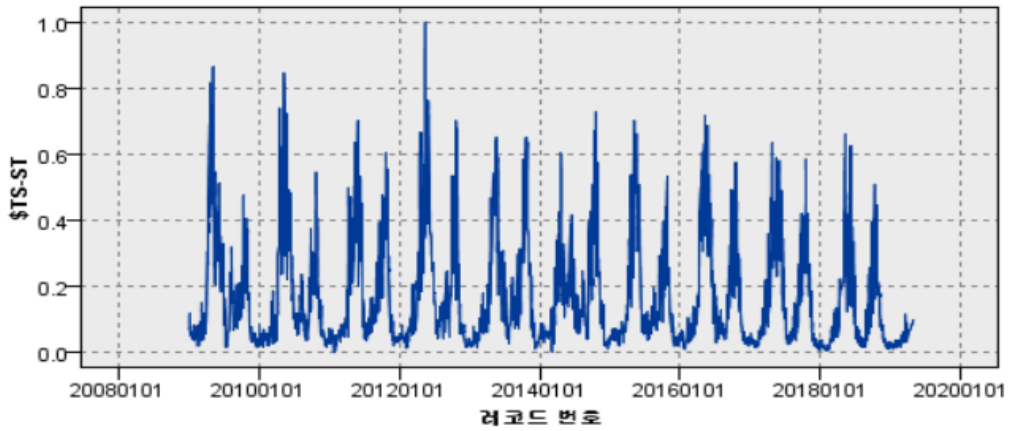
4. 데이터 분석
(1) 시계열 분석

- 일별 예측

1) 2019년 4월까지의 기존 분포



2) 2019년 4월 예측 분포



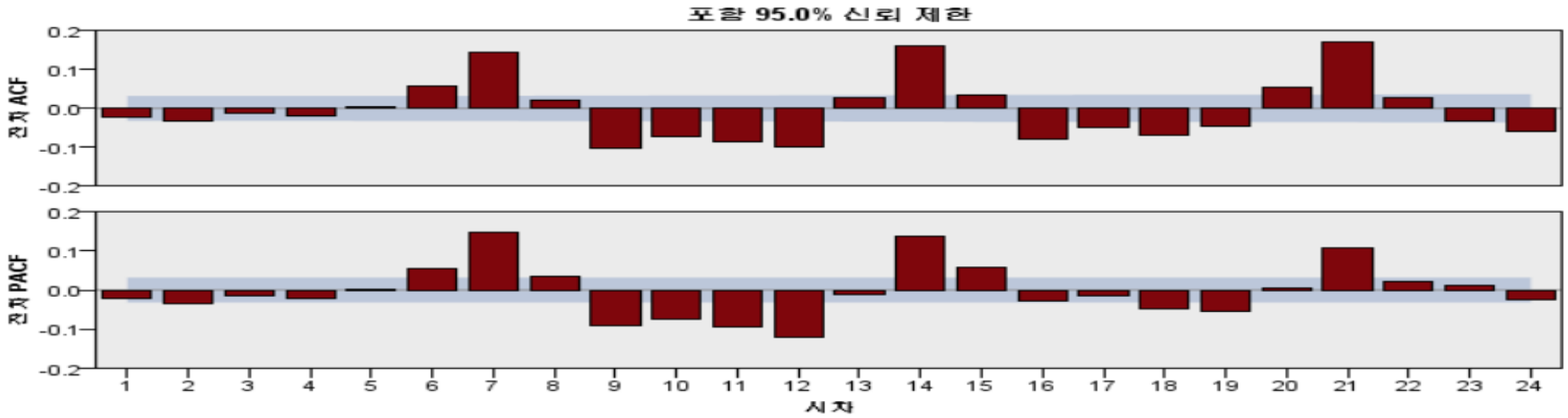
→ 일별 기존 값(ST) / 예측 값($TS-ST)

$TI_TimeLabel	ST
20190328	1808
20190329	3072
20190330	2596
20190331	4471
20190401	1918
20190402	2173
20190403	5671
20190404	3362
20190405	5535
20190406	11537
20190407	32324
20190408	6588
20190409	3485
20190410	2925
20190411	12194
20190412	11149
20190413	39526
20190414	5215
20190415	7007
20190416	7828
20190417	6547
20190418	5689
20190419	8260
20190420	29699
20190421	20205
20190422	4710
20190423	4959
20190424	4418
20190425	12363
20190426	5593
20190427	24366
20190428	18507
20190429	3925
20190430	9415

$TI_TimeLabel	$TS-ST
20190328	2206
20190329	2433
20190330	3566
20190331	4022
20190401	3409
20190402	3311
20190403	2932
20190404	2993
20190405	3294
20190406	3272
20190407	3332
20190408	3392
20190409	3454
20190410	3517
20190411	3582
20190412	3647
20190413	3713
20190414	3781
20190415	3850
20190416	3920
20190417	3992
20190418	4065
20190419	4139
20190420	4215
20190421	4292
20190422	4370
20190423	4450
20190424	4531
20190425	4613
20190426	4698
20190427	4783
20190428	4871
20190429	4960
20190430	5050

4. 데이터 분석
(1) 시계열 분석

3) 예측 결과에 대한 잔차



→ 자기상관함수(ACF)와 편자기상관함수(PACF) 이용하여 오차항들의 자기상관 확인 결과, 대체적으로 파란색 칸을 넘어 자기상관이 있으므로 **적절하지 않은 모형**

4) 정확도 및 적합도 확인

목표	모델	예측변수	Stationary...	R**2	RMSE	MAPE	MAE
ST	ARIMA(0,1,7)	0	0.336	0.419	6,851.162	128.015	4,229.043

4. 데이터 분석
(2) 로지스틱 회귀분석

- 범주형 데이터인 입장객수 등급을 예측

1) 명목형 변수를 Dummy 변수로 변경 : Day, Dust, Event_total, WarningLevel

In [524]:
```
dummies_Day = pd.get_dummies(timeseries_data['Day'])
dummies_dust = pd.get_dummies(timeseries_data['dust'])
dummies_warninglevel = pd.get_dummies(timeseries_data['warninglevel'])
dummies_season = pd.get_dummies(timeseries_data['season'])
```

In [525]:
```
timeseries_data = timeseries_data.join(dummies_Day.add_prefix('Day_'))
timeseries_data = timeseries_data.join(dummies_dust.add_prefix('dust_'))
timeseries_data = timeseries_data.join(dummies_warninglevel.add_prefix('warninglevel_'))
timeseries_data = timeseries_data.join(dummies_season.add_prefix('season_'))
```

In [526]:
```
re_columnlist = ['Day_월', 'Day_화', 'Day_수', 'Day_목', 'Day_금', 'Day_토', 'Day_일'
                ,'dust_초미세먼지', 'dust_미세먼지', 'dust_둘 다 있음', 'dust_좋음'
                ,'warninglevel_경보', 'warninglevel_주의보', 'warninglevel_없음'
                ,'season_봄', 'season_여름', 'season_가을', 'season_겨울'
                ,'event_total'
                ,'rain_mean', 'temper_mean', 'daily_predict'
                ,'ST_level']
```

In [527]:
```
timeseries_data = timeseries_data[re_columnlist]
```

1) 명목형 변수를 Dummy 변수로 변경한 결과

Out[528]:

	Day_월	Day_화	Day_수	Day_목	Day_금	Day_토	Day_일	dust_초미세먼지	dust_미세먼지	dust_둘 다 있음	...	warninglevel_없음	season_봄
1	0	0	0	0	1	0	0	0	0	0	...	1	0
2	0	0	0	0	0	1	0	0	0	0	...	1	0
3	0	0	0	0	0	0	1	0	0	0	...	1	0
4	1	0	0	0	0	0	0	0	0	0	...	1	0
5	0	1	0	0	0	0	0	0	0	0	...	1	0

5 rows × 23 columns

4. 데이터 분석
(2) 로지스틱 회귀분석

2) Train/Test Set을 70%, 30%으로 설정

In [533]:
```
train_test_timeseries_data = timeseries_data[timeseries_data.index < 3742]
month4_timeseries_data = timeseries_data[timeseries_data.index >= 3742]
```

In [534]:
```
X = train_test_timeseries_data[['Day_월', 'Day_화', 'Day_수', 'Day_목', 'Day_금', 'Day_
            ,'dust_초미세먼지', 'dust_미세먼지', 'dust_둘 다 있음', 'dust_좋음'
            ,'warninglevel_경보', 'warninglevel_주의보', 'warninglevel_없음'
            ,'season_봄', 'season_여름', 'season_가을', 'season_겨울'
            ,'event_total'
            ,'rain_mean', 'temper_mean', 'daily_predict']]
Y = train_test_timeseries_data['ST_level']
```

In [535]:
```
X_train, X_test, Y_train, Y_test = train_test_split(X.values, Y.values, test_size=0.30)
```

In [536]:
```
print('X_train의 크기',np.shape(X_train))
print('Y_train의 크기',np.shape(Y_train))
print('X_test의 크기',np.shape(X_test))
print('Y_test의 크기',np.shape(Y_test))
```

```
X_train의 크기 (2618, 22)
Y_train의 크기 (2618,)
X_test의 크기 (1123, 22)
Y_test의 크기 (1123,)
```

4. 데이터 분석
(2) 로지스틱 회귀분석

3) Logistic Regression 모델 설계

In [537]:
```
from sklearn.linear_model import LogisticRegression
```

In [538]:
```
LRC = LogisticRegression()
LRC.fit(X_train, Y_train)
```

```
C:₩Users₩lhj91₩Anaconda3₩lib₩site-packages₩sklearn₩linear_model₩logistic.py:432: Futu
reWarning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to sil
ence this warning.
  FutureWarning)
C:₩Users₩lhj91₩Anaconda3₩lib₩site-packages₩sklearn₩linear_model₩logistic.py:469: Futu
reWarning: Default multi_class will be changed to 'auto' in 0.22. Specify the multi_c
lass option to silence this warning.
  "this warning.", FutureWarning)
```

Out[538]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [539]:
```
predicted = LRC.predict(X_test)
```

In [540]:
```
accuracy = LRC.score(X_test, Y_test)
print("Logistic Regression test file accuracy:" + str(accuracy))
```

```
Logistic Regression test file accuracy:0.7693677649154052
```

→ 정확도: **76.9%**

4) 4월 데이터 예측 결과

In [542]:
```
month4_X = month4_timeseries_data[['Day_월', 'Day_화', 'Day_수', 'Day_목', 'Day_금', 'D
                    ,'dust_초미세먼지', 'dust_미세먼지', 'dust_둘 다 있음', 'dust_좋음
                    ,'warninglevel_경보', 'warninglevel_주의보', 'warninglevel_없음'
                    ,'season_봄', 'season_여름', 'season_가을', 'season_겨울'
                    ,'event_total'
                    ,'rain_mean', 'temper_mean', 'daily_predict']]

month4_Y = month4_timeseries_data['ST_level']
```

In [544]:
```
predicted = LRC.predict(month4_X)

accuracy = LRC.score(month4_X, month4_Y)
print("Logistic Regression test file accuracy:" + str(accuracy))
```

```
Logistic Regression test file accuracy:0.6666666666666666
```

→ 정확도: **66.7%**

→ 문제점 : Train / Test를 구분하여 Logistic Regression 모델링을 만든 경우에는 78%이지만, 실제로 새로운 데이터를(4월) 만났을 때 모델이 잘 맞지 않음

해결 방안 : 보다 의미 있는 변수를 찾아야 할 것으로 판단 및 더 많은 데이터 확보하여 학습 필요함.

1)의사결정나무

- 목표변수가 범주형 → 의사결정나무 중 분류나무(Classification Tree) 이용

- 예측에 사용한 데이터 Attribute

No	Attribute	Attribute 내용	표현방식/단위	역할
1	Date	날짜	연속형	없음
2	Day	요일	명목형	입력
3	ST	일일 입장객수	연속형	없음
4	rain_mean	강수량	연속형	입력
5	temper_mean	기온	연속형	입력
6	event_total	행사의 유무	이분형	입력
7	dust	미세먼지/초미세먼지 유무	명목형	입력
8	warninglevel	미세먼지/초미세먼지 수준	명목형	입력
9	season	계절	명목형	입력
10	level	일일 입장객수 등급	명목형	목표

4. 데이터 분석
(3) 의사결정나무

1) 의사결정나무 – Training set 검증

- **데이터 샘플링**

: random하게 training set 70%, test set 30% 비율로 샘플링

· 샘플링 전 데이터

값	비율	%	빈도
A		76.48	2862
B		13.74	514
C		5.75	215
D		2.91	109
E		1.12	42

· Training set

값	비율	%	빈도
A		76.11	1424
B		14.0	262
C		5.83	109
D		2.78	52
E		1.28	24

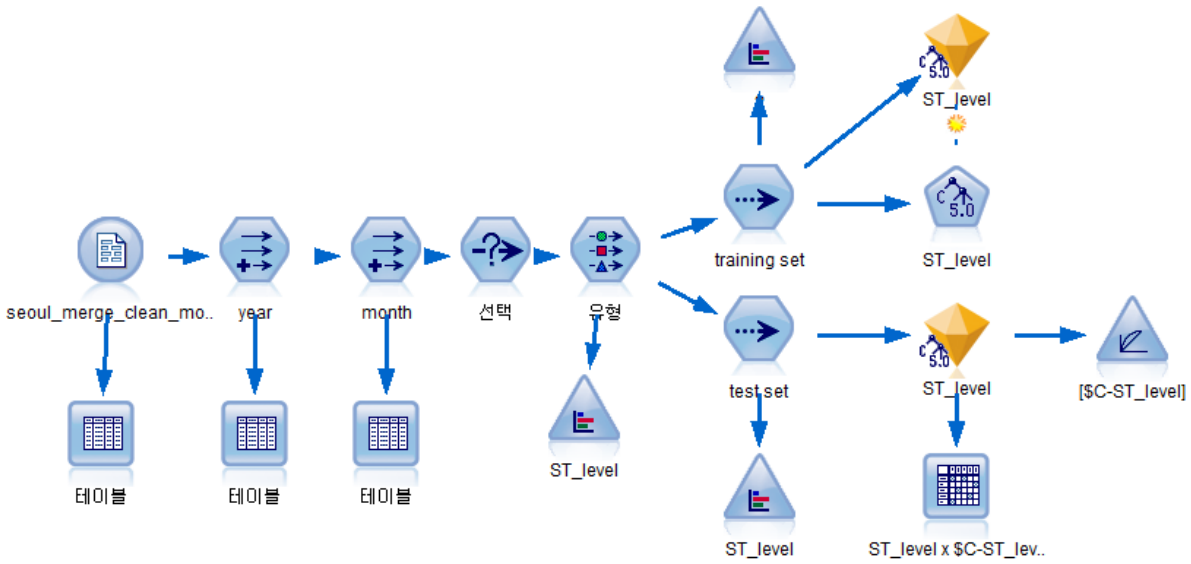
· Test set

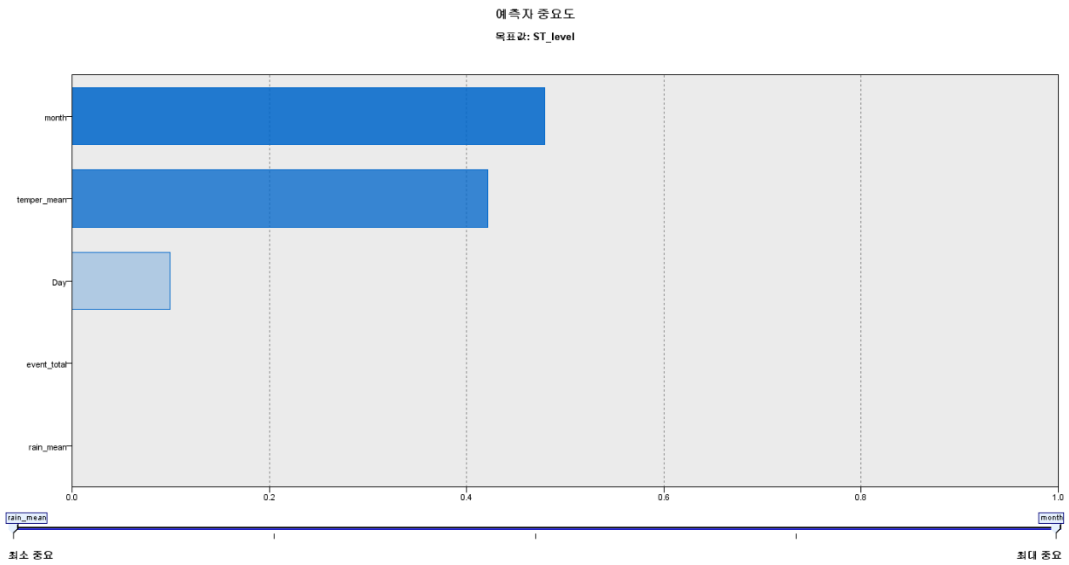
값	비율	%	빈도
A		76.86	1438
B		13.47	252
C		5.67	106
D		3.05	57
E		0.96	18

4. 데이터 분석
(3) 의사결정나무

1) 의사결정나무

(1) C5.0 모델



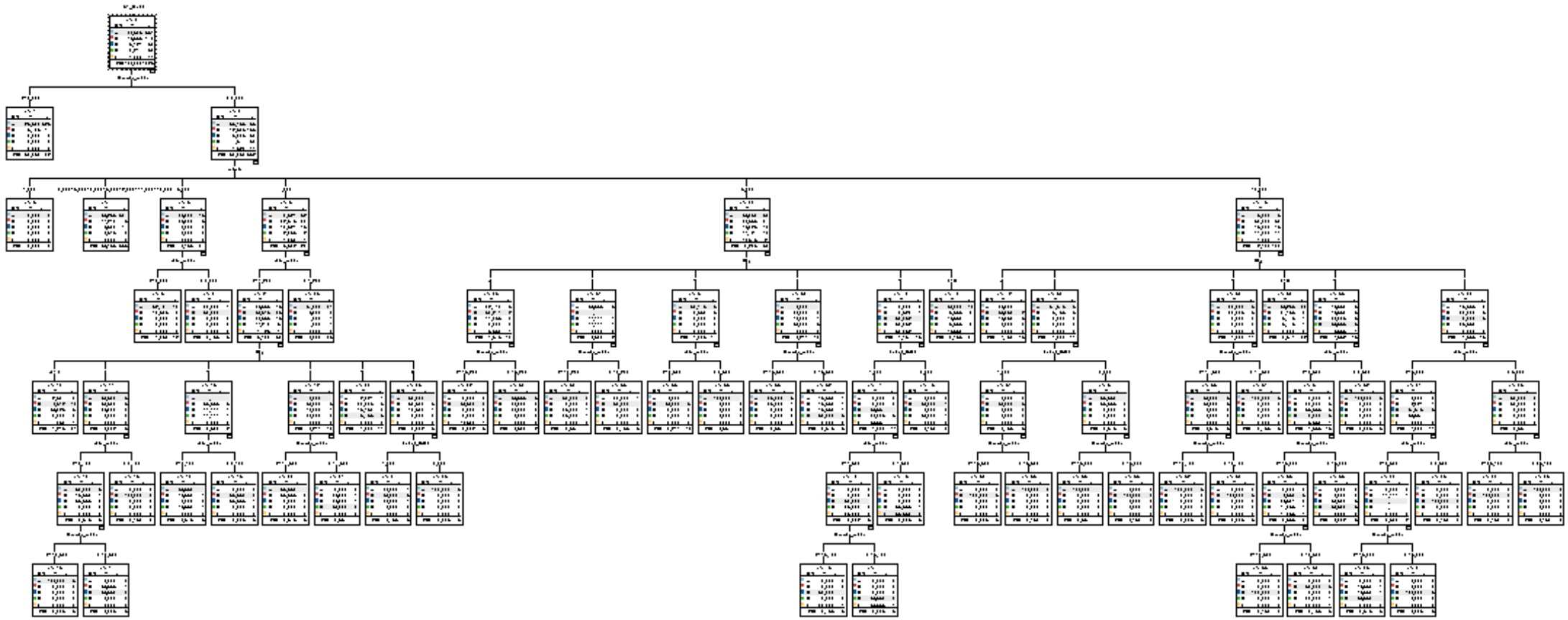


→ month, temper_mean, Day, event_total, rain_mean 변수 유의함

→ 입장객 수 예측에 가장 중요 예측변수: **month**

1) 의사결정나무

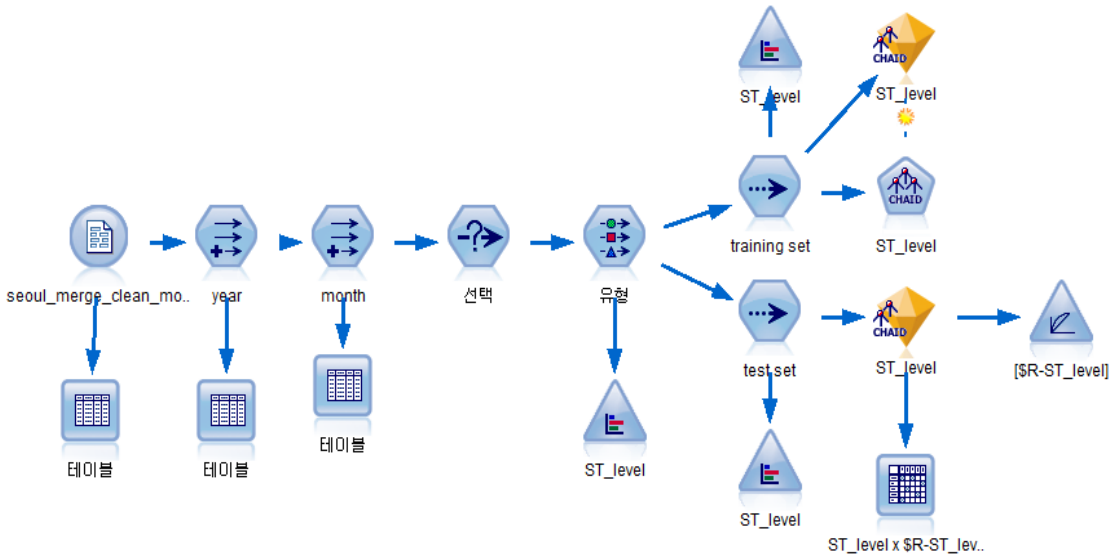
(1) C5.0 모델

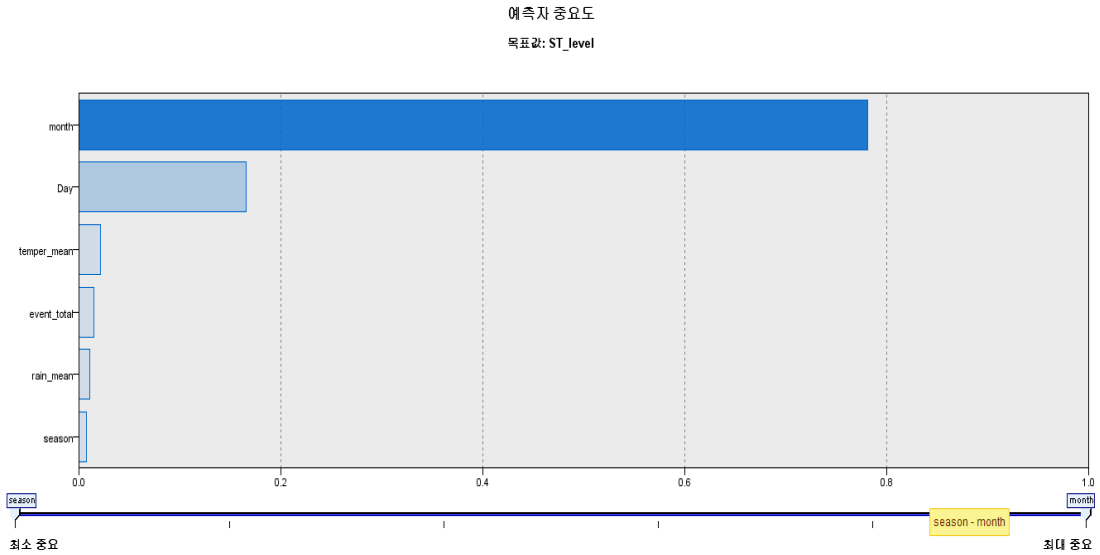


4. 데이터 분석
(3) 의사결정나무

1) 의사결정나무

(2) CHAID 모델





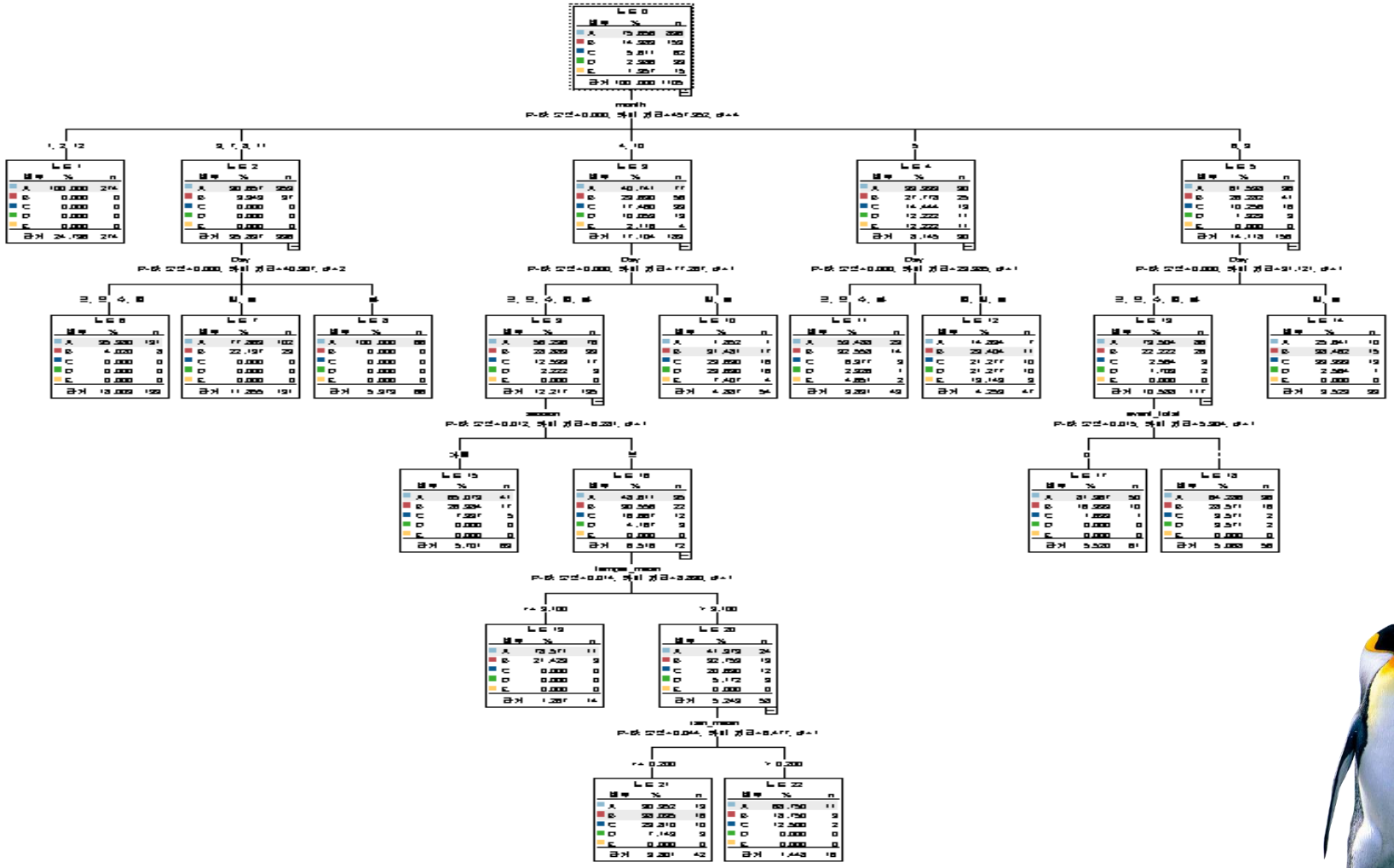
→ month, Day, temper_mean, event_total, rain_mean, season 변수 유의함

→ 입장객수 예측에 제일 중요한 변수: **month**

4. 데이터 분석
(3) 의사결정나무

1) 의사결정나무

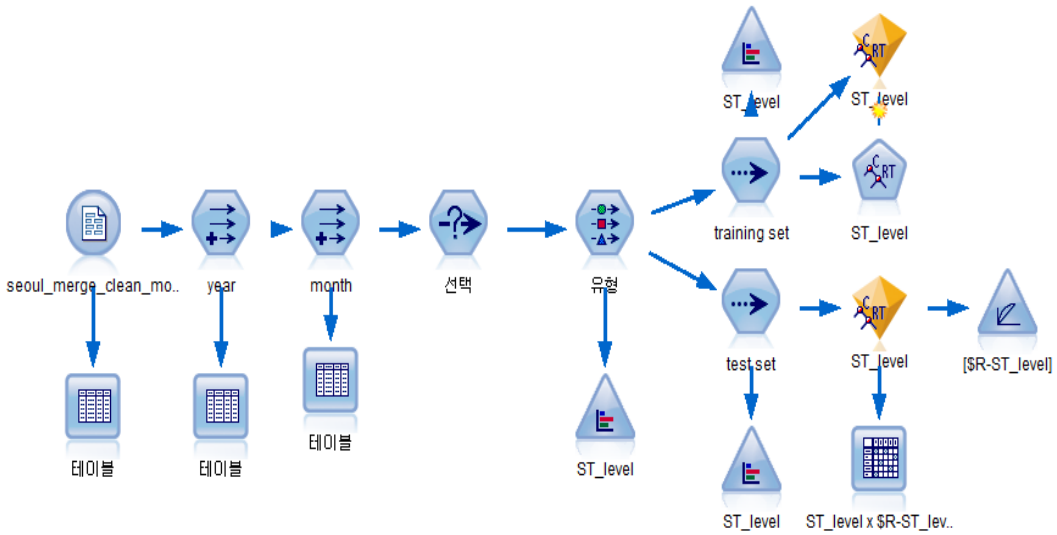
(2) CHAID 모델

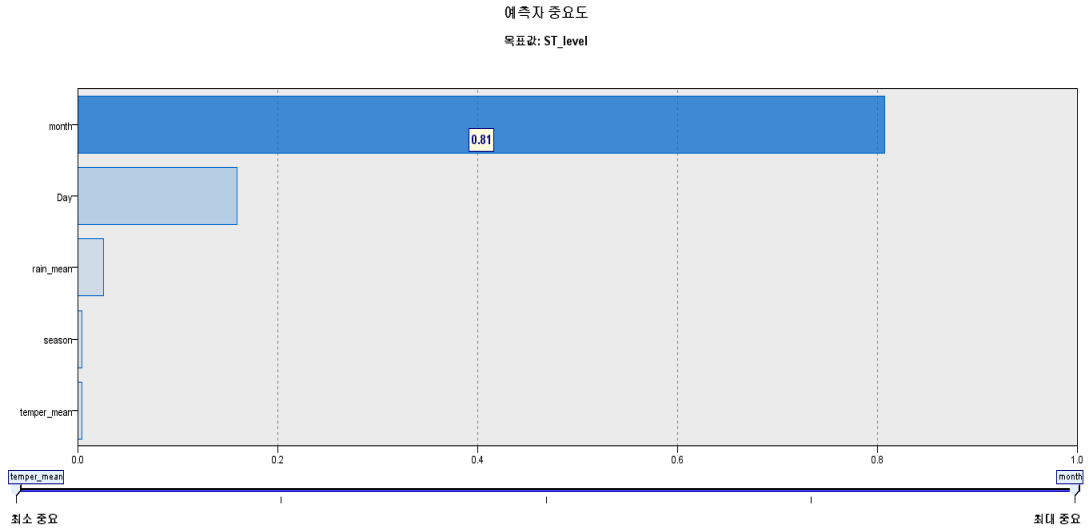


4. 데이터 분석
(3) 의사결정나무

1) 의사결정나무

(3) CART 모델





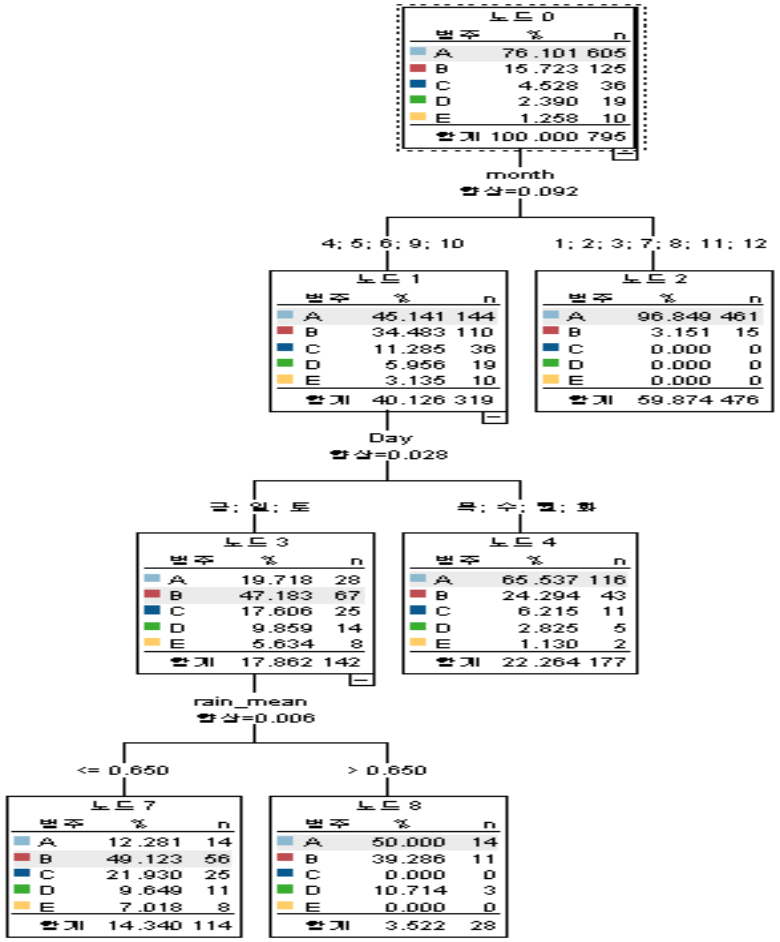
→ month, Day, rain_mean, season, temper_mean 변수 유의함

→ 입장객수 예측에 가장 중요한 변수: **month**

4. 데이터 분석
(3) 의사결정나무

1) 의사결정나무

(3) CART 모델



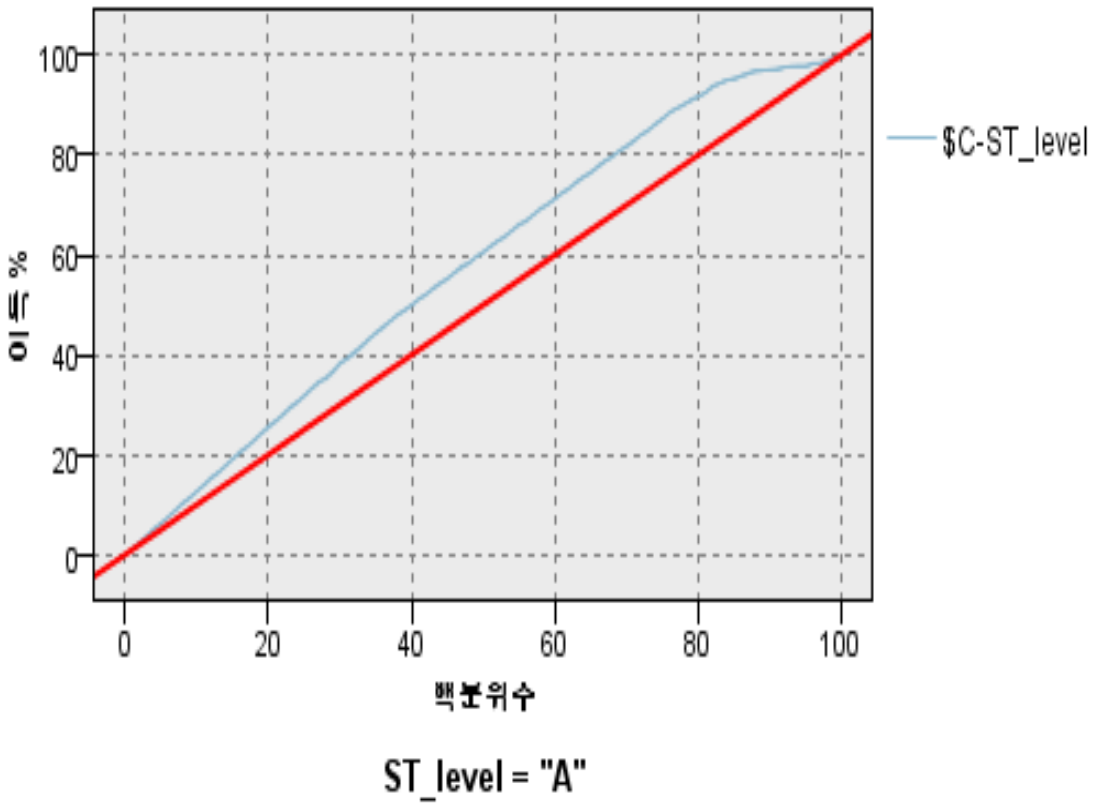
4. 데이터 분석
(3) 의사결정나무

2) C5.0/CHAID/CART 모델 - Test Set 검증

(1) C5.0 모델 교차표 분석 결과: 서울대공원 입장객수에 관한 예측 정확도 80.1%

$C-ST_level

ST_level		A	B	C	D	E	합계
A	빈도	844	26	4	2	0	876
	행 %	96.347	2.968	0.457	0.228	0.000	100
	열 %	85.166	28.571	40.000	5.556	0.000	77.522
B	빈도	99	41	2	6	0	148
	행 %	66.892	27.703	1.351	4.054	0.000	100
	열 %	9.990	45.055	20.000	16.667	0.000	13.097
C	빈도	35	11	3	11	1	61
	행 %	57.377	18.033	4.918	18.033	1.639	100
	열 %	3.532	12.088	30.000	30.556	50.000	5.398
D	빈도	10	9	0	16	0	35
	행 %	28.571	25.714	0.000	45.714	0.000	100
	열 %	1.009	9.890	0.000	44.444	0.000	3.097
E	빈도	3	4	1	1	1	10
	행 %	30.000	40.000	10.000	10.000	10.000	100
	열 %	0.303	4.396	10.000	2.778	50.000	0.885
합계	빈도	991	91	10	36	2	1130
	행 %	87.699	8.053	0.885	3.186	0.177	100
	열 %	100	100	100	100	100	100

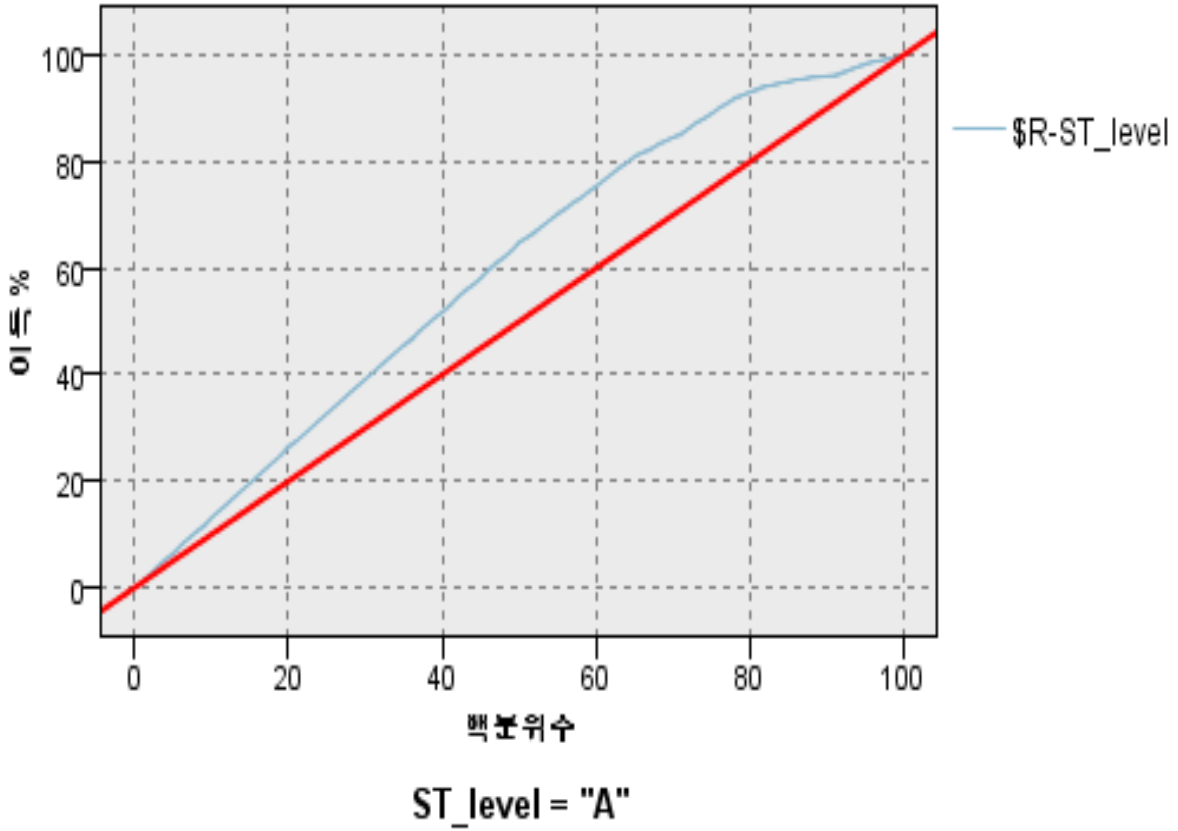


2) C5.0/CHAID/CART 모델 - Test Set 검증

(2) CHAID 교차표 분석결과: 서울대공원 입장객수에 관한 예측 정확도 78.2%

$R-ST_level

ST_level		A	B	합계
A	빈도	803	40	843
	행 %	95.255	4.745	100
	열 %	87.378	21.164	76.083
B	빈도	88	63	151
	행 %	58.278	41.722	100
	열 %	9.576	33.333	13.628
C	빈도	23	46	69
	행 %	33.333	66.667	100
	열 %	2.503	24.339	6.227
D	빈도	3	30	33
	행 %	9.091	90.909	100
	열 %	0.326	15.873	2.978
E	빈도	2	10	12
	행 %	16.667	83.333	100
	열 %	0.218	5.291	1.083
합계	빈도	919	189	1108
	행 %	82.942	17.058	100
	열 %	100	100	100

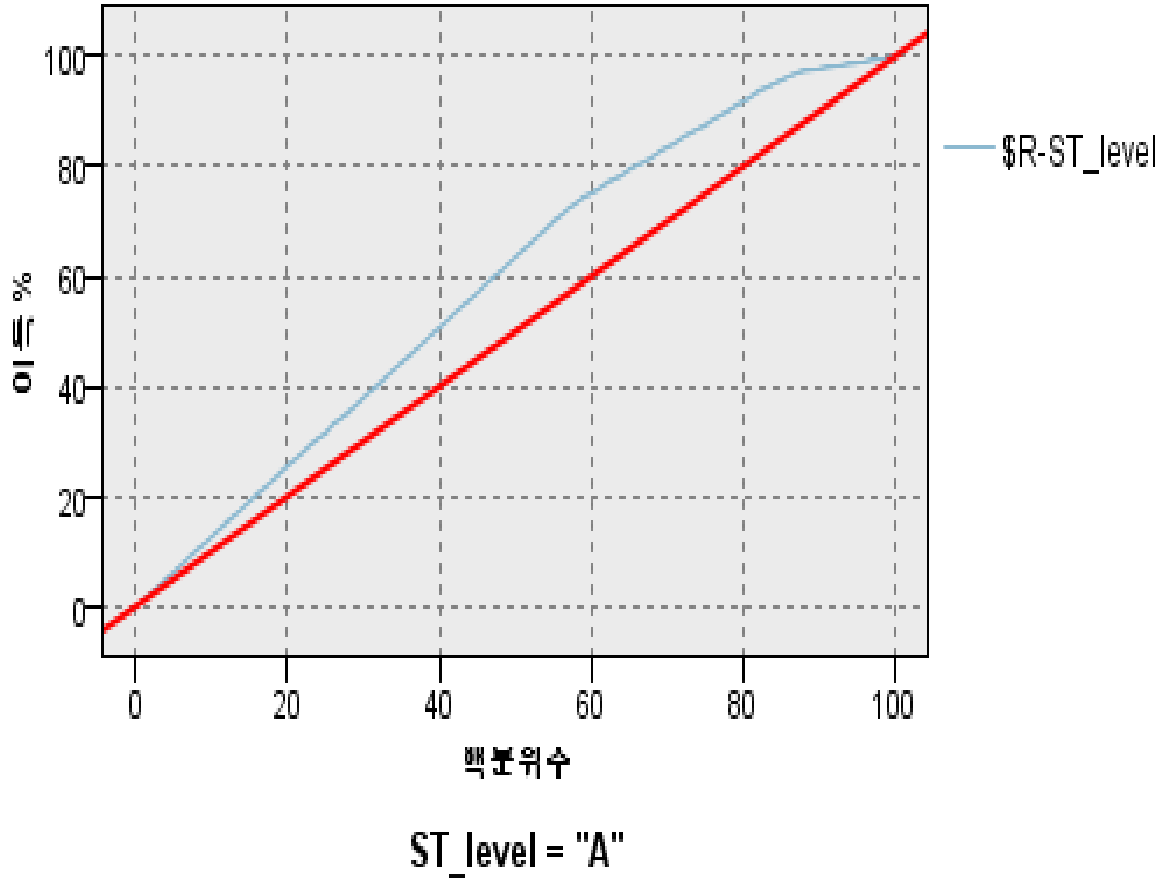


2) C5.0/CHAID/CART 모델 - Test Set 검증

(3) CART 교차표 분석 결과: 서울대공원 입장객수에 관한 예측 정확도 79.8%

$R-ST_level

ST_level		A	B	합계
A	빈도	835	20	855
	행 %	97.661	2.339	100
	열 %	86.439	12.500	75.933
B	빈도	85	63	148
	행 %	57.432	42.568	100
	열 %	8.799	39.375	13.144
C	빈도	29	44	73
	행 %	39.726	60.274	100
	열 %	3.002	27.500	6.483
D	빈도	13	28	41
	행 %	31.707	68.293	100
	열 %	1.346	17.500	3.641
E	빈도	4	5	9
	행 %	44.444	55.556	100
	열 %	0.414	3.125	0.799
합계	빈도	966	160	1126
	행 %	85.790	14.210	100
	열 %	100	100	100



4. 데이터 분석
(4) 시각화

- 로지스틱 회귀분석에서 도출된 2019년 4월의 일일 입장객 수 예측 결과를 Excel로 시각화

1) 예측된 2019년 4월의 일일 입장객 수 등급

일	월	화	수	목	금	토
	4월 1일 09:00~19:00 A	4월 2일 09:00~19:00 A	4월 3일 09:00~19:00 A	4월 4일 09:00~19:00 A	4월 5일 09:00~19:00 A	4월 6일 09:00~19:00 A
4월 7일 09:00~19:00 A	4월 8일 09:00~19:00 A	4월 9일 09:00~19:00 A	4월 10일 09:00~19:00 A	4월 11일 09:00~19:00 A	4월 12일 09:00~19:00 A	4월 13일 09:00~19:00 A
4월 14일 09:00~19:00 A	4월 15일 09:00~19:00 A	4월 16일 09:00~19:00 A	4월 17일 09:00~19:00 A	4월 18일 09:00~19:00 A	4월 19일 09:00~19:00 A	4월 20일 09:00~19:00 A
4월 21일 09:00~19:00 A	4월 22일 09:00~19:00 A	4월 23일 09:00~19:00 A	4월 24일 09:00~19:00 A	4월 25일 09:00~19:00 A	4월 26일 09:00~19:00 A	4월 27일 09:00~19:00 A
4월 28일 09:00~19:00 A	4월 29일 09:00~19:00 A	4월 30일 09:00~19:00 A				

2) 실제 2019년 4월의 일일 입장객 수 등급

일	월	화	수	목	금	토
	4월 1일 09:00~19:00 A	4월 2일 09:00~19:00 A	4월 3일 09:00~19:00 A	4월 4일 09:00~19:00 A	4월 5일 09:00~19:00 A	4월 6일 09:00~19:00 B
4월 7일 09:00~19:00 D	4월 8일 09:00~19:00 A	4월 9일 09:00~19:00 A	4월 10일 09:00~19:00 A	4월 11일 09:00~19:00 B	4월 12일 09:00~19:00 B	4월 13일 09:00~19:00 D
4월 14일 09:00~19:00 A	4월 15일 09:00~19:00 A	4월 16일 09:00~19:00 A	4월 17일 09:00~19:00 A	4월 18일 09:00~19:00 A	4월 19일 09:00~19:00 A	4월 20일 09:00~19:00 C
4월 21일 09:00~19:00 B	4월 22일 09:00~19:00 A	4월 23일 09:00~19:00 A	4월 24일 09:00~19:00 A	4월 25일 09:00~19:00 B	4월 26일 09:00~19:00 A	4월 27일 09:00~19:00 C
4월 28일 09:00~19:00 B	4월 29일 09:00~19:00 A	4월 30일 09:00~19:00 A				

→ 정확도: **66.67%**

4. 데이터 분석
(4) 시각화

- 의사결정나무에서 도출된 2019년 4월의 일일 입장객 수 예측 결과를 Excel로 시각화

1) 예측된 2019년 4월의 일일 입장객 수 등급

일	월	화	수	목	금	토
	4월 1일 09:00~19:00 A	4월 2일 09:00~19:00 A	4월 3일 09:00~19:00 A	4월 4일 09:00~19:00 A	4월 5일 09:00~19:00 B	4월 6일 09:00~19:00 B
4월 7일 09:00~19:00 D	4월 8일 09:00~19:00 A	4월 9일 09:00~19:00 A	4월 10일 09:00~19:00 A	4월 11일 09:00~19:00 A	4월 12일 09:00~19:00 B	4월 13일 09:00~19:00 B
4월 14일 09:00~19:00 D	4월 15일 09:00~19:00 A	4월 16일 09:00~19:00 B	4월 17일 09:00~19:00 A	4월 18일 09:00~19:00 A	4월 19일 09:00~19:00 B	4월 20일 09:00~19:00 B
4월 21일 09:00~19:00 D	4월 22일 09:00~19:00 D	4월 23일 09:00~19:00 A	4월 24일 09:00~19:00 A	4월 25일 09:00~19:00 A	4월 26일 09:00~19:00 A	4월 27일 09:00~19:00 B
4월 28일 09:00~19:00 D	4월 29일 09:00~19:00 A	4월 30일 09:00~19:00 A				

2) 실제 2019년 4월의 일일 입장객 수 등급

일	월	화	수	목	금	토
	4월 1일 09:00~19:00 A	4월 2일 09:00~19:00 A	4월 3일 09:00~19:00 A	4월 4일 09:00~19:00 A	4월 5일 09:00~19:00 A	4월 6일 09:00~19:00 B
4월 7일 09:00~19:00 D	4월 8일 09:00~19:00 A	4월 9일 09:00~19:00 A	4월 10일 09:00~19:00 A	4월 11일 09:00~19:00 B	4월 12일 09:00~19:00 B	4월 13일 09:00~19:00 D
4월 14일 09:00~19:00 A	4월 15일 09:00~19:00 A	4월 16일 09:00~19:00 A	4월 17일 09:00~19:00 A	4월 18일 09:00~19:00 A	4월 19일 09:00~19:00 A	4월 20일 09:00~19:00 C
4월 21일 09:00~19:00 B	4월 22일 09:00~19:00 A	4월 23일 09:00~19:00 A	4월 24일 09:00~19:00 A	4월 25일 09:00~19:00 B	4월 26일 09:00~19:00 A	4월 27일 09:00~19:00 C
4월 28일 09:00~19:00 B	4월 29일 09:00~19:00 A	4월 30일 09:00~19:00 A				

→ 정확도: **53.33%**

5. 결론

시계열		의사결정나무			로지스틱회귀
Sationary R Squared	MAPE	C5.0	CHAID	CART	Accuracy
0.336	128.02	80.10%	78.20%	79.80%	66.70%

서울대공원 입장객 수 증감에 영향을 미치는 요인으로는 '몇 월'이냐가 제일 큰 것으로 파악.
그 다음으로는 온도, 요일, 행사 유무, 강수량 순.

∴ 서울대공원 측에서는 이 세 요인을 중점으로 하여 입장객 수 예측 & 예측된 입장객 수에 대비한 적절한 수의 직원 배치 전략 수립 가능

6. 느낀 점

Q&A

감사합니다